Architecting, Designing and Building a Secure Information Technology Infrastructure, a case study

GSEC Practical Version 1.4b

Author: John David Johnston
Date: 24 August 2003

Table of Contents

Table of Figures

Abstract

This case study follows the building of an Information Technology Infrastructure with an integrated Security Architecture. Describing this project as a case study indicates that this is a practical discussion not a theoretically one. This paper follows the process from concept to implementation. It shows the results of a pre-project analysis, follows the project through completion, examining the steps along the way. It concludes with a discussion of the post-project analysis and a comparison of the results from the two analyses. The paper will discuss what was done, why it was done and how it was accomplished. Conclusions are drawn on the relative success of the project.

Although the project involves building an entire ITI this paper focuses on the security aspects. It will cover the high-level architecture and some of the low-level implementation details.

The paper is written from the perspective of a System Administrator who has an intense interest in information security. It has two main parts. Part I introduces the terminology used and describes the design approach, laying a foundation for the second part. Part II uses that foundation to help describe the project.

The necessity of implementing information security will not be argued. The assumption here is that security is required. Given our current global Information Technology environment, one should not build an Information Technology Infrastructure without designing in a Security Architecture.[1] This paper addresses how to approach and successfully accomplish that task. We start with a discussion of terminology.

Part I—Project Foundation

Before we start the project we need to establish an operational framework. Some of the concepts mentioned here may be basic and perhaps widely known; they are included for clarity and completeness.

Terminology

Information is defined as "a signal or character (as in a communication system or computer) representing data". [2] I define Information Technology as the tools and techniques employed to store, transfer and process electronic data. The tools are the hardware and software components. Techniques are the methods used to manipulate those tools to perform a required task. An infrastructure is, defined again by Webster, the "underlying foundation or basic framework (as of a system or organization)". Taken together they form Information Technology Infrastructure (ITI).

Security means protecting the Confidentiality, Integrity and Availability of our Information.[3] "The Architecture of a system refers to how it is designed and how the components of the system are connected to, and operate with, each other."[4] This combination gives us Security Artchitecture (SA).

Putting all the terms together we have Information Technology Infrastructure Security Architecture (ITISA) or how the basic framework of the tools and techniques used within an enterprise to process electronic data are design to provide us with confidentiality,

[1] If you are not convinced that security is a necessity just perform a google search and you will find many convincing arguments.
[2] Webster's Universal Encyclopedic Dictionary. Barnes & Nobel, 2002. 945.
[3] Cole, Eric. Fossen, Jason. Northcutt, Stephen. Pomeranz, Hal. SANS Security Essentials with CISSP CBK Version 2.1 Volume One Sans Press, 2003
[4] Newton, Harry Newton's Telecom Dictionary. San Francisco: CMP Books, 2003. 70.

Integrity and availability of our electronic information. The ITISA is a key concept used through out this paper. How does the ITISA fit within the broader topic of enterprise security?

Enterprise Security Architecture

Securing an enterprise presents a complex problem. Many of the discussions reviewed approach the problem from its highest level, the Enterprise Security Architecture (ESA).

> "A successful security architecture combines a heterogeneous combination of policies and leading practices, technology, and a sound education and awareness program."[5] "The objective of enterprise security architecture is to provide the ***conceptual design*** of the network security infrastructure, related security mechanisms, and related security policies and procedures".[6]

Such a high level view is extremely important and should not be ignored. However, solving a problem the size of securing information requires that it be broken into more manageable sections then tackled individually. Chang Boon Tee takes that approach in his discussion of building a secure Internet Data Center.[7] This paper will take a similar approach by exploring a lower level of the enterprise security hierarchy. To reach the target level our discussion requires us to explore the concept of modularity.

Modular Security Architecture[8]

A modular view of security architecture treats each component individually. This gives us two concepts, architectures within architectures and the coexistence of peer architectures. We can now look at security in two important ways, from a hierarchical view and from an independent component view. With a hierarchical view we can see the underlying architectures. A horizontal view helps us understand the interrelationship between peer component architectures. An example might be helpful.

The Personal Computer (PC) has a hierarchical and modular architecture. The hardware architecture has a CPU, input/output, and storage devices as components. Its software architecture has an operating system and applications as its components. If we drill down a level we uncover the microprocessor and operating system architectures. This modularity provides the designer with the ability to architect and develop the components independently. Similar components with different architectures become interchangeable. The ability to swap components requires an additional ingredient, compatible interfaces. In

[5] Angelo, Scott M. "Security Architecture Model Component Overview." 27 Nov. 2001. 3, 15 URL: http://www.sans.org/rr/securitybasics/architecture.php. (5 Apr. 2003)
[6] Arconati, Nick. "One Approach to Enterprise Security Architecture." 14 Mar. 2002. 1, 2 URL: http://www.sans.org/rr/policy/approach.php. (5 Apr. 2003)
[7] Tee, Chang Boon. "Building a secure Internet Data Center Network Architecture." URL: http://www.sans.org/rr/paper.php?id=73 (20 Apr. 2003)
[8] Modules can be either components or elements of components. This will be relevant later when we discuss the project architecture and design.

the PC world interfaces have become standardized to make them compatible. PCI, SCSI, and USB are just a few examples.

The Enterprise Security Architecture can also be viewed as hierarchical and modular. The two main components are technological and non-technological. Scott Angelo, referenced above, distinguishes the two as "technical controls and non-technical controls. "…the technical controls, [are] the various hardware and software elements required to secure the overall infrastructure." and "…the non-technical controls, [are the] processes and procedures." Nick Arconati, also referenced above, list "Documentation, Services and Technology, with Technology being Intrusion Detection Systems, Firewalls, & Host-based Protection."

The "technical controls" and the "technology" are components of the ITI. How those components are designed and how they interact is the Security Architecture. Applying the layered modular approach we can see that the Information Technology Infrastructure Security Architecture is a layered modular component of the Enterprise Security Architecture. To interact with the parent ESA, the ITISA, like the PC modules, must have interfaces.

ITISA Interfaces

Wendell Odom states that, "Standardized interfaces among layers facilitates modular engineering."[9] We can also say, modular designs require standardized interfaces. The ITISA communicating with other modules of the ESA is really technology communicating with humans. The ITISA communicates through logs, reports, alerts and programs. The logs are raw data, reports are analyses and summations of raw data, and alerts are real time messages sent to humans indicating an event took place that requires attention. The programs are used to electronically enforce specified policies. These are the interfaces that facilitate modular design.

As we will see, there are benefits to a layered modular design.

Benefits of Using a Layered Modular Design

Using a layered modular approaches is considered fundamental to system design. When discussing the design of network topologies Priscilla Oppenheimer recommends a "hierarchical network design …technique for designing scalable campus and enterprise networks using a layered, modular model."[10] She later states, "Modularity lets you keep each design element simple and easy to understand." Additional benefits of using the hierarchical model also include, cost savings, easy network grow and improved fault isolation.[11]

[9] Odom, Wendell. Cisco CCNA Exam #640-507 Certification Guide. Indianapolis: Cisco Press, 2000. 78.
[10] Oppenheimer, Priscilla. Top-down Network Design. Indianapolis: Macmillan Technical Publishing, 1999. 121, 123.
[11] Teare, Diane. Designing Cisco Networks Indianapolis: Cisco Press 1999. 90.

Using the layered modular approach to implementing security offers the same benefits. Each module can be easily understood. When a fault arises it can be isolated to the offending module. Modules can be designed and developed independently with a focus on its specialized function. Modules can be replicated and installed as required to provide scalability.

With the benefits of modularity in mind we can examine the modules or components of the ITISA.[12]

The Components of the ITISA

The main components of the ITISA are the same as those listed as layers in the strategy of defense in depth. They are network, host, applications and information.[13] We will look at each component closer with a definition, the types of attacks against its vulnerabilities and some defenses.

The network is a "collection of computers, printers, routers, switches, and other devices that are able to communicate with each other over some transmission medium."[14] The vulnerabilities that can exist in a network are unused open ports, incorrect firewall rulesets, insecure code and absence of adequate perimeter defense. There are two main types of attacks, denial of service and break-ins. Defenses typically deployed are Network Intrusion Detection, firewalls, and vulnerability scanning followed by system hardening.

A host is a"…computer connected to a network".[15] Vulnerabilities include open unused ports, weak passwords, unused services and unpatched software. Attacks are typically against the user accounts, the operating system, services, and ports. Defenses that can be used are strong passwords, host based intrusion detection, file integrity checkers, single service systems and regular patching routines.

An application is a "software program that carries out some useful task."[16] The vulnerabilities are lack of error checking and susceptibility to buffer overflows. The main type of attacks against an application involves exploiting buffer overflows. Unfortunately the typical System Administrator has little or no control of the selection or development of applications. Keeping the application patched and restricting access are the two main things that can be done.

Information is defined again as, "a signal or character (as in a communication system or computer) representing data". The vulnerabilities are repositories that are insecure and the lack of access control. The usual attacks include stealing and corrupting data.

---

[12] I tend to use module, component and sometimes element interchangeable. I hope this does not confuse anyone to much. Modules can be components and visa versa. Please consider the context.
[13] Cole, Eric. Fossen, Jason. Northcutt, Stephen. Pomeranz, Hal. SANS Security Essentials with CISSP CBK Version 2.1 Volume One Sans Press, 2003. 294.
[14] Teare, Diane. Designing Cisco Networks Indianapolis: Cisco Press 1999. 737.
[15] "Dictionary.com" URL: http://dictionary.reference.com/search?q=host. (25 Apr. 2003).
[16] Newton, Harry Newton's Telecom Dictionary. San Francisco: CMP Books, 2003. 67.

Defenses involve encryption, file protection mechanisms, compartmentalized data storage and access control lists.

The threat to security is multi-tiered and must be countered with a multi-level defense. The ITISA provides the framework for that multi-level defense by following the strategy of defense in depth.

Building the ITISA using the concepts of Defense In Depth

Defense in depth is a widely accepted strategy for implementing security. The strategy builds successive layers of defensive measures at strategic points between the potential attacker and the valued assets. The valued assets, in the digital world, are in the form of electronic information. That electronic information can represent various types of data from credit card information to intellectual property. To the computer it is just 0's and 1's. Once those special sets of 0's and 1's, the corporate jewels, are identified they must be protected from the attacker. By employing defense in depth the attacker must penetrate each layer successfully and undetected before reaching the corporate jewels. The ITISA is the defense in depth gauntlet standing between the attacker and the corporate jewels.

The ITISA has been identified as an independent module of the ESA. There are some situations where one may want to take advantage of that independence.

Reasons for building Independent ITISA modules

It may be necessary to implement an ITISA independently of an ESA for a number of reasons.

- A directive may have been issued to implement security because of an attack or the fear of a pending attack and there is no time to develop a full security policy and a complete set of procedures.

- A company may only have enough resources to implement the technology portion of security.

- IT personnel are building a new ITI. There is no one to address the non-technical portions of the ESA. They recognize that it is much easier to implement security at this point then attempting to retrofit it after the network is in production.

- An astute System Administrator gets hired, recognizes that there is no security in place, major vulnerabilities exist, and realizes who has to clean up if there is a successful attack.

- An ITISA is required to be developed in parallel with and maybe slight ahead of the other components of the ESA.

All of these situations can benefits from the modular approach to security that building an independent ITISA can provide. As you will soon see, our project fits one of the scenarios. Before we move on to the project let us summarize.

Summary

We have a description of the terminology that will be used and have explored the modular design methodology, listing its benefits. The ITISA was introduced as a module of the ESA, its modules examined and an explanation provided for how they will be used. We have a description of how the ITISA interfaces with the other modules of the ESA. We reviewed the strategy of defense in depth and how important it is in setting up security defenses. Finally we listed some reasons why this modular approach makes sense and provided some situations where it may be developed independently. With this foundation we can move on to project.

Part II—The project

This part of the paper describes the project. The main goal of this project is to implement an Information Technology Infrastructure Security Architecture (ITISA) for an unnamed company following the principles and methods described above.

Background

The name selected for this project is gauntlet, as in “run the gauntlet” not “throw down the gauntlet”, a very important distinction.[17] It would be foolhardy to challenge attackers to break into your system. Gauntlet seems appropriate since we want the attacker to face multiple obstacles before getting to the prize. It also helps to remind us that the best security defense, like a gauntlet, can be successfully negotiated, at least Clint Eastwood was able to in the movie of the same name. Our desire is that the attacker will become too bloodied and bruised to continue and move on to easier pickings.

It was necessary to examine the existing ITI. This provided important insights into its use and clues to future requirements useful to design process. It uncovered legacy applications and requirements for future support.

To gather as much information as possible a questionnaire was developed, a preliminary security analysis was performed, a physical inventory conducted and a network diagram produced. A summary of relevant information follows.

The company performs IT consulting and provides contract services consisting of IT infrastructure design and implementation, IT project management, UNIX/Linux System and Network Administration and Information Security. This is a one-person outfit but the intention is to grow over the next few years. Standard popular office applications are used

[17] For an explanation of the difference genealogies in the meaning of gauntlet see http://www.crh.noaa.gov/library/Grammar/Gauntlet.html. (30 Mar. 2003)

running on a Windows platform. There is a set of lab equipment currently used for training. The web site and pop email server is colocated at an ISP. A second ISP provides Internet access. The project is to be completed by the end of summer 2003. Internet access, mail and web, cannot be interrupted for more than one hour a day.

The new ITI must be capable of supporting multiple users in a secure fashion. There may be a future need for storing client data. Company data must be secured from other users on the network. Secure communication is required including wireless and, later, remote Internet access and dial-in. The equipment budget cannot exceed $500.

A backup strategy and first level disaster recovery plan needs to be developed. A full business continuation plan will come later.

Pre-project Network Architecture

The network is linked to the Internet through a WAN modem to the IPS's router. The WAN modem is connected via Ethernet to a 486 box with two network interface cards running Debian Linux and performing firewall task, routing, and many to one NATing. That firewall system is connected to an Ethernet switch. The switch has ports connected to three desktop systems and one laptop. Two of the desktops are multi-boot systems. One boots into either RedHat Linux 8.0 or Solaris 9. The other can boot into Windows 2000 Server or Linux 8.0. The third desktop runs Windows 98, has a printer attached and is connected via a hot sync cable to a PDA running Palm v4. The laptop can boot into RedHat Linux 8.0, Windows 2000 professional or Solaris 9 X86. See Figure 1

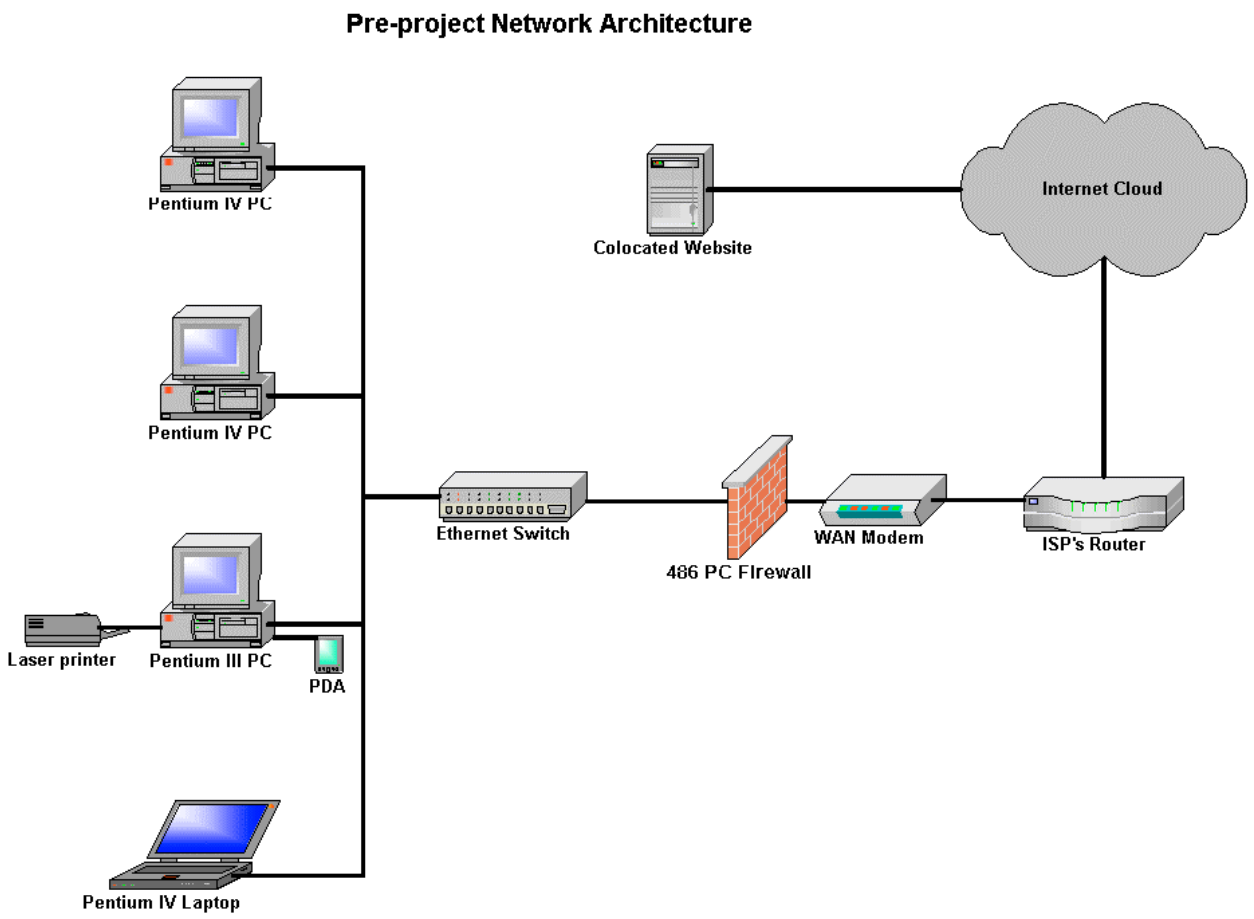

Figure 1: Pre-project Network Architecture

The results of the physical inventory are listed in Figure 2.

Device	Function	OS	Version
Systems			
486 DX/2	Firewall	Debian	2.2.18pre21
Pentium III PC	Gen. Desktop	Windows	98
Pentium IV Laptop	Mobility	Windows	2K Pro
		Redhat Linux	8
		Solaris	8 X86
Pentium 133Mhz Laptop	Not Currently Used	Redhat Linux	7.3
Pentium IV PC	Laboratory Workstation	Redhat Linux	8
		Windows	2K Pro
Pentium IV PC	Laboratory Workstation	Redhat Linux	8
		Windows	2K SVR
SparcSTATION 2	Not Currently Used	Solaris	7
Network Devices			
WAN Modem	ISP Connection		
802.11b PC card	Wireless connectivity		
802 11b PC card	Wireless connectivity		
5 Port switch	Ethernet Switching		
4 Port Hub	Non-switched connections		
Other Devices			
KVM switch	Console Access		
BW Laser Printer	Printing		

Figure 2: Device List

Pre-project Security Architecture

The pre-project security architecture has one main component, a firewall. There is also an anti-virus checker running on the desktop PC. There is no security policy. An important part of analyzing the ITI was to perform a pre-project security assessment[18].

Pre-project Security Assessment

The purpose of the security assessment was to determine the state of security and uncover vulnerabilities, addressing any that needed immediate attention. Anything not considered an emergency would be resolved with the new architecture. The security assessment collected data that would be used for comparison at the end of the project. The preliminary assessment also provided an opportunity to become more familiar with the various security analysis tools.

The assessment used the same components of the ITISA identified above, the network, the hosts, the applications and the data. [19] The assessment followed the same path an

[18] The assessment was conducted because there is no security policy to use as the basis for an audit.
[19] As it turns out, application would require a different approach. More on this later.

attacker would probably use when attempting to perpetrate a break in. The attacker would conduct reconnaissance to gain as much information as possible about the target. That would include performing whois queries to collect IP addresses, domain names and contact information. Scans would be conducted to determine operating system information for the perimeter device. A port scan would be launched to find open ports and the services running on them. Research would be done to see if known vulnerabilities existed and exploits available. If the perimeter device is a firewall, utilities are run to see if packets can be forwarded through the firewall to the internal network. If the attacker can get into the internal network a network scan is conducted.

The network scan is done to map the topology of the network and determine what kinds of devices are available. After obtaining a description of the network, attempts would be made to break into vulnerable systems. If access is gained all kinds of mischief can be perpetrated, Trojan horses are left, back doors opened and attempts made to elevate privilege. If the attacker gains root access, game over, they own your system.

That was a very brief description of the break in process. It is far more involved and technical then that. There are a couple of useful references that can provide an in depth understanding of the break in process. One is Hackers Beware, by Eric Cole.[20] The other is Hacking Exposed, by Stuart McClure, Joel Scambray and George Kurtz.[21]

Below is a summary of the assessment starting with the network.

Network

Since the company's web site and email address point to the colocation site and not the site being assessed that kind of reconnaissance would gain little information about the local site.

Cheops-ng was run to ID the OS of the firewall.[22] Cheops-ng reported that the OS could be Linux 2.1.19 – 2.2.20. The firewall was scanned with nessus and it reported a number of open ports, some possible vulnerabilities and suggested methods of fixing them.[23]

Following this firewalk was run to determine if any of those ports were being forwarded through the firewall, none were.[24]

I felt it was prudent to close the open ports on the firewall immediately. Closing the ports was easy. It took 5 minutes to modify the inetd.conf file commenting out the offending services and restarting inetd, then finding the startup scripts that initiated the rpc and postfix services, stopping and disabling them so they would not restart on reboot. [25]

---

[20] Cole, Eric. Hackers Beware. Indianapolis: New Riders Publishing, 2002
[21] McClure, Stuart. Scambray, Joel. Kurtz, George. Hacking Exposed. Berkeley: McGraw Hill/Osborn, 2003.
[22] "Cheops-ng." URL: http://cheops-ng.sourceforge.net/index.php. (25 May 2003).
[23] "Nessus." URL: http://www.nessus.org/. (25 May 2003).
[24] "Firewalk." URL: http://www.packetfactory.net/firewalk/. (25 May 2003).
[25] Postfix was installed but never fully implemented so it was never used.

Then testing with a reboot and running nessus again to make certain they were still closed after the reboot.

Further investigation of the firewall revealed that the OS was Debian 2.2.18pre21 released August 14th, 2000 (current release is 3.0). The release had never been patched so it contained all of the vulnerabilities listed in the alerts at the Debian site.[26]

The deb package ipmasq v3.0.19 developed by Brian Basset was used to create the firewall rules and IP masquerading using Ipchains, a stateless firewall protocol, as the traffic filter.[27] Because of the age of the firewall implementation and lack of maintenance, examining the exploits and vulnerabilities, defenses and tools would be pointless. Beside a simple upgrade to the latest version would eliminate the majority of those vulnerabilities.

Host

The assessment continued inside the firewall as if an attacker had successfully breached the firewall. The network was first scanned with nessus. Each system that could multiboot was booted into each OS and scanned in turn. Nessus revealed that numerous ports were open and needed to be evaluated and addressed.

John The Ripper was run against the Linux password files.[28] It took five days to crack the user's password and had not cracked the root password after ten days when it was stopped.

Applications

Most of the applications used are resident on the Windows 98 system. Since this system will obviously have to be replaced the applications used on the new system will have to be evaluated as they are implemented. The security will improve just by upgrading to a new operating system.

Data

All data on the system is stored without file permissions, in non-encrypted form, primarily on a win98. If an attacker gets to this point the company jewels are there for the taking.

Recovery/Availability

There is no documented backup and recovery methodology. Backups are performed on an irregular basis to a zip drive for recently changed information. A disk-to-disk backup is done for major portion of the data. Backups are not stored offsite.

[26] "Security Information." 19 Aug. 2003. URL: http://www.debian.org/security/. (22 Aug. 2003).
[27] Basset, Brian. URL: http://packages.debian.org/stable/net/ipmasq.html. (10 Aug 2003).
[28] "John the Ripper." URL: http://www.openwall.com/john/. (1 Jun. 2003).

The pre-project assessment determined that a complete redesign was necessary. Those problems considered emergency were fixed to provide a more secure environment until project completion.

The next step in the project was the development of a functional specification. The specification provided a clear understanding of the goal and what is to be built. Its approval made certain that there was agreement on the project and provided a document to follow. A functional specification can be written without a full-blown security policy but it does require a security framework. What is the difference?

Security Policy versus a Security Framework

Security Policy can be defined by the combining the following two definitions. Security was defined earlier as "Confidentiality, Integrity and Availability". The SANS Security Policy Project states, "A policy is typically a document that outlines specific requirements or rules that must be met."[29] The security framework is the operations parameters for implementing the technical portions of the security infrastructure. That framework can be suggested by the implementer but needs approval before starting the project. That process should be quick, easy and not delay the project. The security framework can later be expanded and incorporated into the security policy.

The Security Architecture provides us with a security framework.

Architecture

This section describes the Security Architecture. It provides a high level description of the main components and sub-components that make up the architecture, their functions and relationships. A better understanding will be gained if we look at how the SA relates to the ITI. The components that make up the information technology infrastructure are shown in figure 3.

[29] "The SANS Security Policy Project." URL: http://www.sans.org/resources/policies/#name. (24 Aug. 2003).

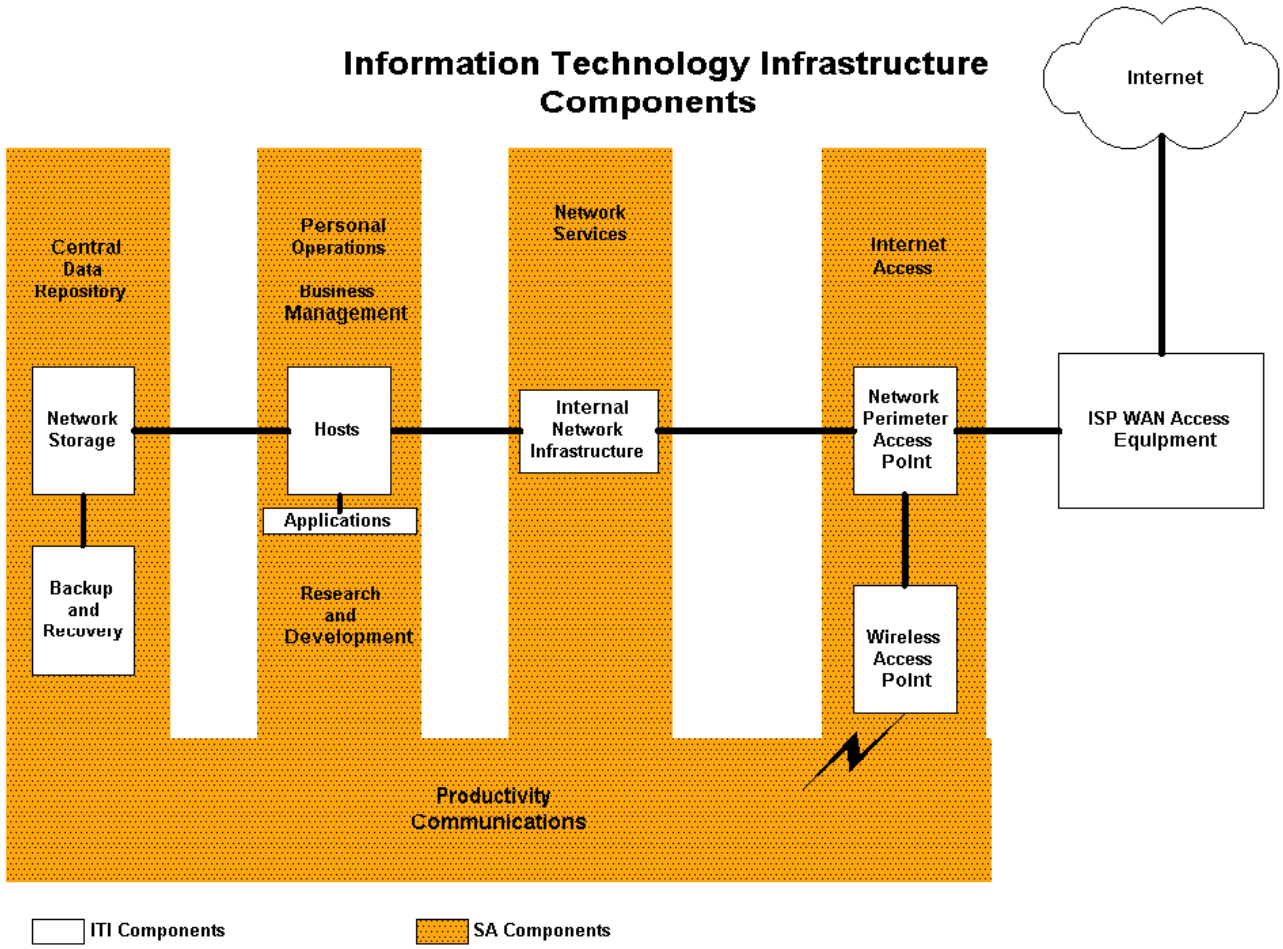

Figure 3: Information Technology Infrastructure Components

Figure 4 shows the security components superimposed on the ITI.

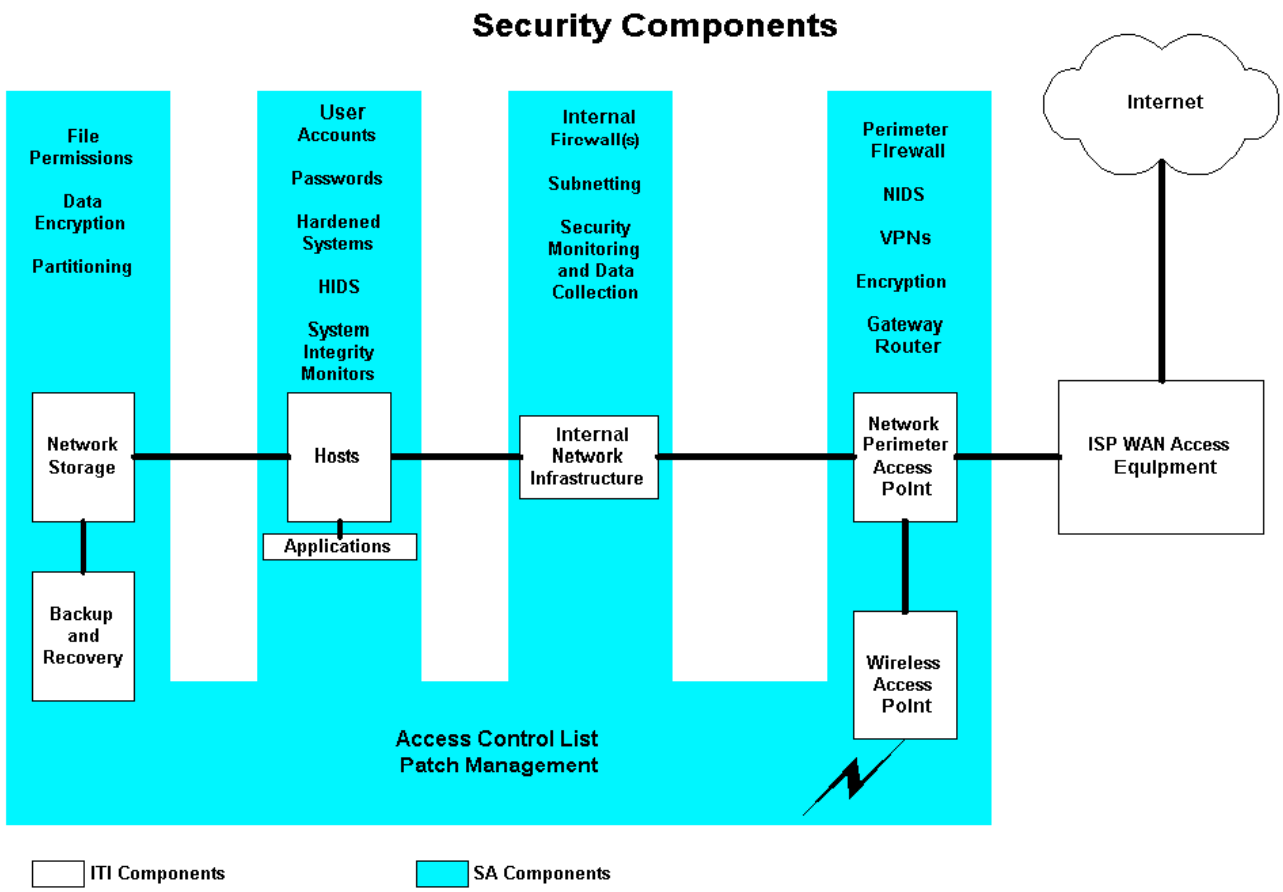

Figure 4: Security Components

There are four main components, the perimeter defense, the network infrastructure protection, the host defenses and the data security mechanisms. The perimeter defense sits at the edge of the internal network and protects it against unauthorized access. The network infrastructure segregates the network into manageable and isolated areas and prevents unauthorized access between subnets. It also provides various services, like monitoring, that track suspicious events happening on the internal network. The host defenses protect individual systems and the applications they run. Data security protects data in transit and data stored on disk to provide the requisite confidentiality, integrity and availability.

Perimeter Defense

The perimeter defense contains three security sub-components, a Transparent Firewall Bridge (TFB), a Security Gateway (SGW) and a Wireless Access Point (WAP). The TFB sits between the Internet and SGW. It monitors traffic to the SGW and acts as a Network Intrusion Detection System (NIDS). The SGW is a firewall performing packet filtering and NATing. It filters unauthorized traffic from entering the internal network and NATs private internal addresses to the Internet address of the SGW. This allows authorized access, via protocol selection, to the Internet. The WAP also does packet filtering allowing

authenticated access from the wireless interface through the SGW to the internal network. It provides an encrypted channel for the wireless access.

Network Infrastructure Protection

The network infrastructure protection provides subnetting and internal firewalling. It also provides central log services and secured DNS. Subnetting is the mechanism used to split a network into segments. Those segments are attached to the SGW and are subject to packet filtering rules. The log server provides a secure backup for system logs. DNS provides name resolution service and does not allow name resolution of the internal network to pass beyond the network perimeter.

Host Defense

The host defense includes Host Intrusion Detection Systems (HIDS), authentication through user names and password requirements, and system hardening. HIDS tracks and detects unauthorized modification to system data. Authentication prevents unauthorized access to the system. Hardening is a process of closing unused ports preventing access to the system through those ports.

Data Security

Data security employs disk partitioning, file permissions, and data encryption. Disk partitioning splits disk into separate entities with each provided with its own access controls. File permission keep individual files and directory protected from unauthorized use. Encryption changes a file to an unreadable state, which can only be change back by those who have the correct key.

The described components working together form the architecture that provides the layers of defense. The elements used to implement the architecture constitute the design.

Design

This section discusses design issues, the design approach and provides details of the elements used to build the architecture. The major design issues included a restrictive budget, scalability functionality versus protection and the share amount of vulnerabilities. The budget issue is addressed first because of its fundamental impact on the design.

Budget constraints meant that there were limited funds to buy commercial security products. It became clear early on that the open source community would have to be a key resource in the design. Using open source requires a different mindset. There is no vendor to call and threaten; there are no maintenance contracts. Anyone using open source must be reasonable self-sufficient, that is, there is no one obligated by compensation to assist you. This can be frustrating at times but not totally bad. The really good part is that there are many people who have attempted to do what you are trying to do and willing to share their knowledge and experiences on the Internet. Another positive

point is that the source code is published and out there for all to examine and rake the developer over the coals if something is not correct or, if they so chose, make important contributions. This results in some high quality and very useful software. Open source was used wherever possible to address the budget issue.

If growth is planned then scalability is important. The security infrastructure must be design for future growth. Once again the benefits of modularity offers the solution. Being able to swap out or replicate modules supplies the needed flexibility to meet the demands of growth.

Another issue was functionality versus protection. Security could not be so cumbersome and overbearing that it interfered with functionality and reduced productivity, yet it still had to provide a reasonable amount of protection. Every attempt was made to design the security into the infrastructure in such a way that it was practically transparent to the authorized user but a genuine nuisance to the attacker. Those who use the system will have to be the final judges of whether this was successful.

The last major issue was the shear amount of vulnerabilities existing in the wild, especially when building a multi-platform IT infrastructure. It became obvious early on that one cannot protect against each and every individual type of attack but most defend against classes of attacks. It made sense to use the same defense in depth philosophy to classify attacks. Following this with a global design decisions to protect those major areas. Also, by selecting the areas must vulnerable presenting the highest risk, spend the most effort there. This was the design approach. The first part of the designed was the security components of the perimeter.

Perimeter Defense

The three components of the perimeter defense are the Transparent Firewall Bridge (TFB), the Security Gateway (SGW) and the Wireless Access Point (WAP).

Transparent Firewall Bridge

The transparent firewall bridge uses Debian as the operating system with a kernel that supports bridging. Snort is used to provide intrusion detection. One interface is attached to the WAN modem. The second is attached to the SGW. The third is attached to an Ethernet switch.

Security Gateway

The SGW is a Pentium system running OpenBSD 3.3. OpenBSD was selected because of that organization's approach to security and their record.[30] The system has two Ethernet interfaces. One has an Internet accessible address and is attached to the bridge. The other is connected to the Ethernet switch and configured with four different subnet

[30] "Security." URL: http://www.openbsd.org/security.html. (1 Apr. 2003).

addresses corresponding to the four internal subnets. Packet Filter (PF), built into the kernel, is used for implementing firewall rule sets.[31]

Wireless Access Point

The WAP is a Pentium III laptop running OpenBSD 3.3. It has two network interfaces. One interface is connected to the switch attached to the Security Gateway. The second interface is an 802.11b PC card configured to use IPsec to supply an encrypted interface for wireless clients. It also doubles as an authentication server using a PF utility named authpf. A squid proxy is installed for web access and to handle SSH connections.[32]

Network Infrastructure Protection

Host Intrusion detection on the OpenBSD system is performed with mtree a built in facility that provides similar functionally to Tripwire.[33] It creates hashes of files listed in a file specification. The hashes can be checked regularly and compared to the initial hashes to determine if the files have been modified and when. Consult the man pages for information on using mtree. Tripwire is used on the Debian system. The OpenBSD system also provides the ability to modify a file's attributes to immutable. By setting the system security level to 2 all users including root are prevented from modifying those files.

All accounts that allow access to the system have passwords. All default accounts not being used have been removed.

Data Security

Encryption

Selected information is encrypted. Data encryption is done with Gnu Privacy Guard.[34]

Backup and Recovery

The backup procedure involves backing up the selected data to a central disk, then compressing and encrypting it. A copy is put on removable media and a copy is sent offsite to an online storage facility. The four main data elements backed up are: 1) Operating systems 2) applications, 3) local configurations and 4) the user data. Each element requires it own procedure. The operating systems and applications CDs are copied with originals kept off-site. A copy of the local configurations are cut to a CD and sent with them. A second copy of the configurations is sent to online off-site storage along with the user data and copies of the log file. This results in having complete sets of crucial information in multiple places.

---

[31] Holland, Nick. "PF: The OpenBSD Packet Filter." 29 Jun. 2003 URL: http://www.openbsd.org/faq/pf/index.html (11 Jul. 2003)
[32] Pearson, Oskar. "Squid, A User's Guide." URL: http://squid-docs.sourceforge.net/latest/html/book1.html. (30 July 2003).
[33] "Tripwire." URL: http://www.tripwire.org/. (10 Aug. 2003).
[34] "Gnu Privacy Guard." URL: http://www.gnupg.org/. (10 Aug. 2003).

The recovery method depends on the severity level of the disaster. Lost files can be recovered from local disk backups. When a data disk becomes unavailable for whatever reason the data can be recovered from removable media. If the entire facility is destroyed the data at the off-site storage can be restored once the facility is replaced.

The above description shows how the selected components are designed into the architecture. Now we can take a look at how it was built.

The Build

This section describes the build process in detail and will cover installation, configuration and testing. To help understand how components are configured I have included some of the configuration files in Appendix A. **First Caveat,** The configuration files are included for your review. There is no guarantee that they will work. You are encouraged to develop and test your own. **Note, IP addresses shown are not the real ones**. This was done for obviously reasons, to provide protection for the company. Well on to the build.

A diagram of the new network infrastructure is shown in figure 5 so it can be referred to when reading the description below.

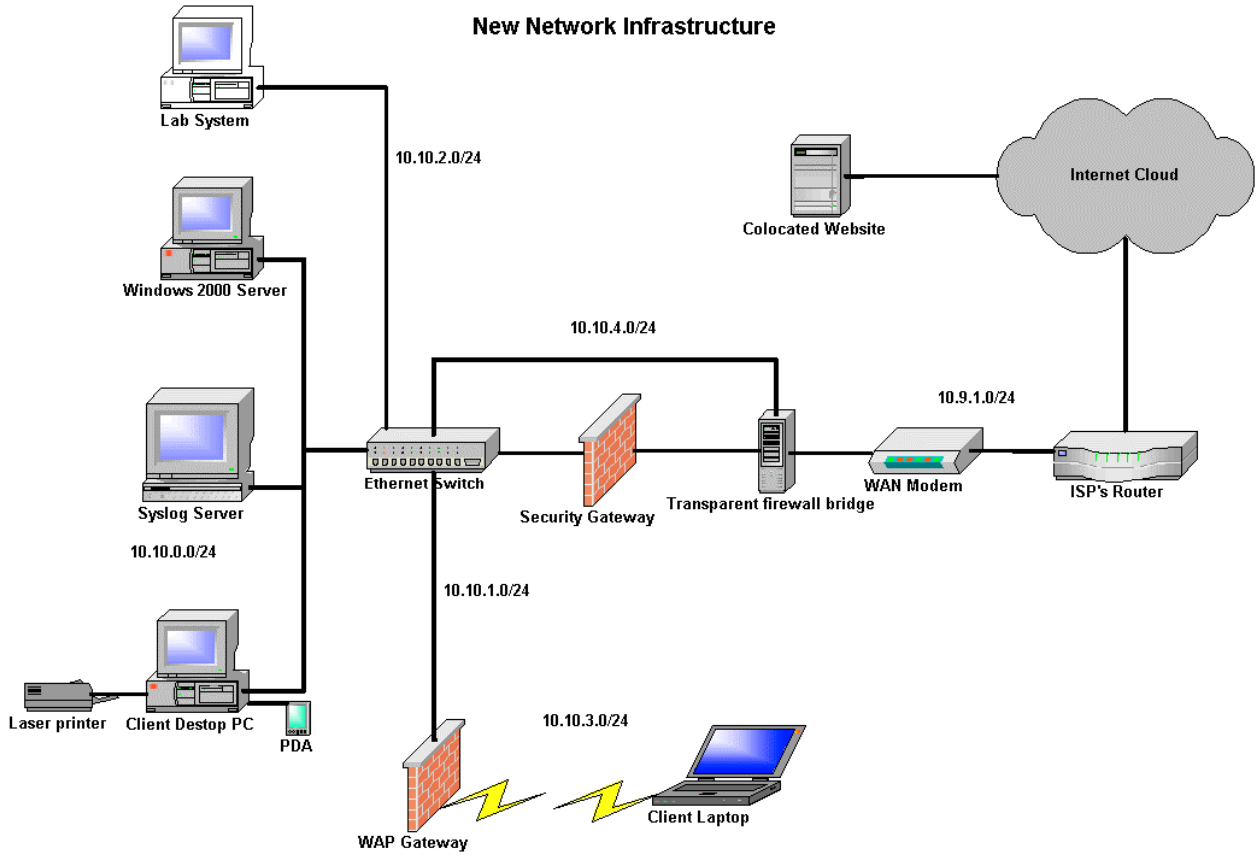

Figure 5: New Network Infrastructure

The approach taken to build this security architecture was to start at the most vulnerable point, the perimeter, and work in.

Perimeter

The main component of perimeter defense is the Security Gateway (SGW). This system has an Internet accessible IP address and blocks or passes traffic to and from the internal network according to a specified set of rules. It uses Packet Filter (PF) as the filtering mechanism. PF is controlled by the pfctl utility and reads /etc/pf.conf by default. There is a "scrub" operation performed on all packets entering the firewall. Scrub reassembles all fragmented packets before further processing. This helps prevent certain kinds of attacks that use invalid TCP flags. The NAT function changes the source IP addresses of all packets coming from the internal network going out to the Internet. It uses a default deny directive so that all authorized traffic must be specifically allowed. An antispoof directive prevents spoofing of the IP address on the external interface.[35]

The PF configuration controls what IP traffic and protocols are permitted to go where. Ping requests with their replies can travel freely within the internal network but not to the external network from the private addresses. It does allow ping from the gateway to the ISP's router. Pings are not allowed in. It passes SSH traffic between the internal subnets and to the shell account at the ISP. IPsec traffic is permitted to pass throughout the internal network including traffic from the wireless network to the Windows 2000 server. DNS traffic can pass from all internal networks to the DNS service on the SGW. The SGW is the only system allowed to send DNS queries externally. No DNS queries are allowed in. A further discussion of DNS appears below. Syslog traffic can pass from any internal system to the syslog server.

Testing was done one rule at a time by using pfctl to alternately turn PF on and off while attempting to send the subject traffic through. Ethereal was used to capture and verify the protocol of the packets.[36] Appendix A shows the pf.conf configuration file.

The other critical component of the perimeter defense is the Wireless Access Point (WAP). OpenBSD 3.3 was installed on the Pentium III laptop and it was able to recognize the Cisco Aironet 350 802.11b wireless card immediately. 802.11b has well known vulnerabilities.[37] IPsec was configured on the wireless interface to overcome the security inadequacies by establishing an encrypted channel for client laptops. The Orinoco card on the client laptop was checked to see if it recognized the Cisco card before proceeding, it did. Configuring IPsec on OpenBSD 3.3 requires the creation of two configuration files isakmpd.conf and isakmpd.policy in the /etc/isakmpd directory. The initial configuration for

[35] Holland, Nick. "PF: The OpenBSD Packet Filter." URL: http://www.openbsd.org/faq/pf/index.html. (11 Jul. 2003).
[36] "Ethereal." URL: http://www.ethereal.com/. (27 Mar. 2003).
[37] Stubblefield, Adam. Ioannidis, John. Rubin, Aviel D. "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP" Rev 2 21 Aug 2001.
URL: http://www.cs.rice.edu/%7Eastubble/wep/wep_attack.pdf

IPsec uses a Pre-shared Secret Key (PSK). Consult the man pages on isakmpd, isakmpd.policy and isakmpd.conf for additional information on configuring isakmpd.

The WAP is a firewall, which also uses PF. It allows the necessary isakmpd protocol packets through to establish and maintain the IPsec encrypted communications. It also restricts wireless logins to SSH over that encrypted channel.

Access to the internal network with SSH and Web access to the Internet is handled with the squid proxy service installed on the WAP. The client systems must be configured to use squid. On the Windows system the web browser is configured manually with the proxy information, the IP address and the proxy port. Squid also has the ability to proxy SSH traffic.[38] PuTTY for windows can be configured to use the squid proxy by adding the IP address and port number of the proxy system and the IP address of the target system. The destination SSH port must be set to 443 so the squid can make the connection. Instruction for configuring PuTTY can be found on their web site.[39] The configuration directive below must be added to the /etc/sshd_config file on the target system enabling it to listen on port 443.[40]

```
"ListenAddress 10.10.1.1:443"
```

The browser configuration on the Linux system is done the same way as it was on the Windows system described above. OpenSSH is configured differently. An additional program called corkscrew is needed to allow SSH to use the proxy.[41] The user creates a configuration file that includes the following line.

```
"ProxyCommand <path to corkscrew> <proxy IP address> <proxy port> %h %p".
```

I did not use ~/.ssh/config, this file is read automatically when SSH is invoked. That would interfere with the SSH authpf login discussed below.

A Windows 2000 server was installed and the user's home directory was moved to it and shared. The server was configured to use IPsec with the same PSK as the WAP and the Laptop. The "User scenario for wireless access", procedure below explains the steps the user takes to gain access to the internal network and the Internet.

IPsec is implemented with ISAKMPD and is used to transfer encrypted data on 802.11b interface. PF is used for rulesets. The authpf utility of PF is a non-interactive login shell used to modify the PF rule sets when the user authenticates on login.

---

[38] Attica. Re: "ssh through squid proxy" URL: http://www.derkeiler.com/Mailing-Lists/securityfocus/Secure_Shell/2003-03/0059.html (30 Jul. 2003).
[39] "PuTTY Documentation Page." URL: http://www.chiark.greenend.org.uk/~sgtatham/putty/docs.html. (15 Aug. 2003).
[40] Norris, Greg. "Re: SSH through squid proxy." 15 Mar. 2003.URL: http://www.securityfocus.com/archive/121/315169 (30 Jul. 2003)
[41] Padgett, Pat. "Corkscrew." URL: http://www.agroman.net/corkscrew/. (30 Jul. 2003).

User scenario for wireless access

When the user's laptop is booted the WAP verifies the PSK and establishes an encrypted IPsec channel over the 802.11b interfaces. The user is now allowed to access the WAP using SSH over that encrypted channel. This is enforced by the PF rulesets. The user logs into the WAP and is authenticated; this session must remain open. After authentication, the authpf utility adds rulesets. Those new rulesets allow the user to establish SSH connections to selected internal systems, send web request to the squid proxy and permit access to the user's shared home directory on the Windows 2000 server.

All traffic passing beyond the WAP must first travel over the encrypted IPsec channel. If it is web or SSH traffic the first hop is encrypted to the proxy then on to its final destination. If the user is accessing a Windows share, that communication is encrypted end to end. Below is a sample of packets captured by ethereal and played back with tcpdump.[42] It shows the encrypted traffic flowing between the two wireless interfaces.

```
12:32:17.271774 10.10.3.5 > 10.10.3.2: ESP(spi=0x8b88b390,seq=0x1)
12:32:17.275519 10.10.3.2 > 10.10.3.5: ESP(spi=0x7d508a86,seq=0x1) (DF)
12:32:17.275619 10.10.3.5 > 10.10.3.2: ESP(spi=0x8b88b390,seq=0x2)
12:32:17.297284 10.10.3.2 > 10.10.3.5: ESP(spi=0x7d508a86,seq=0x2) (DF)
12:32:17.298302 10.10.3.5 > 10.10.3.2: ESP(spi=0x8b88b390,seq=0x3)
12:32:17.298635 10.10.3.5 > 10.10.3.2: ESP(spi=0x8b88b390,seq=0x4)
```

Figure 6 shows the paths taken by the SSH and share traffic.

[42] "Tcpdump." URL: http://tcpdump.org. (20 Mar. 2003).

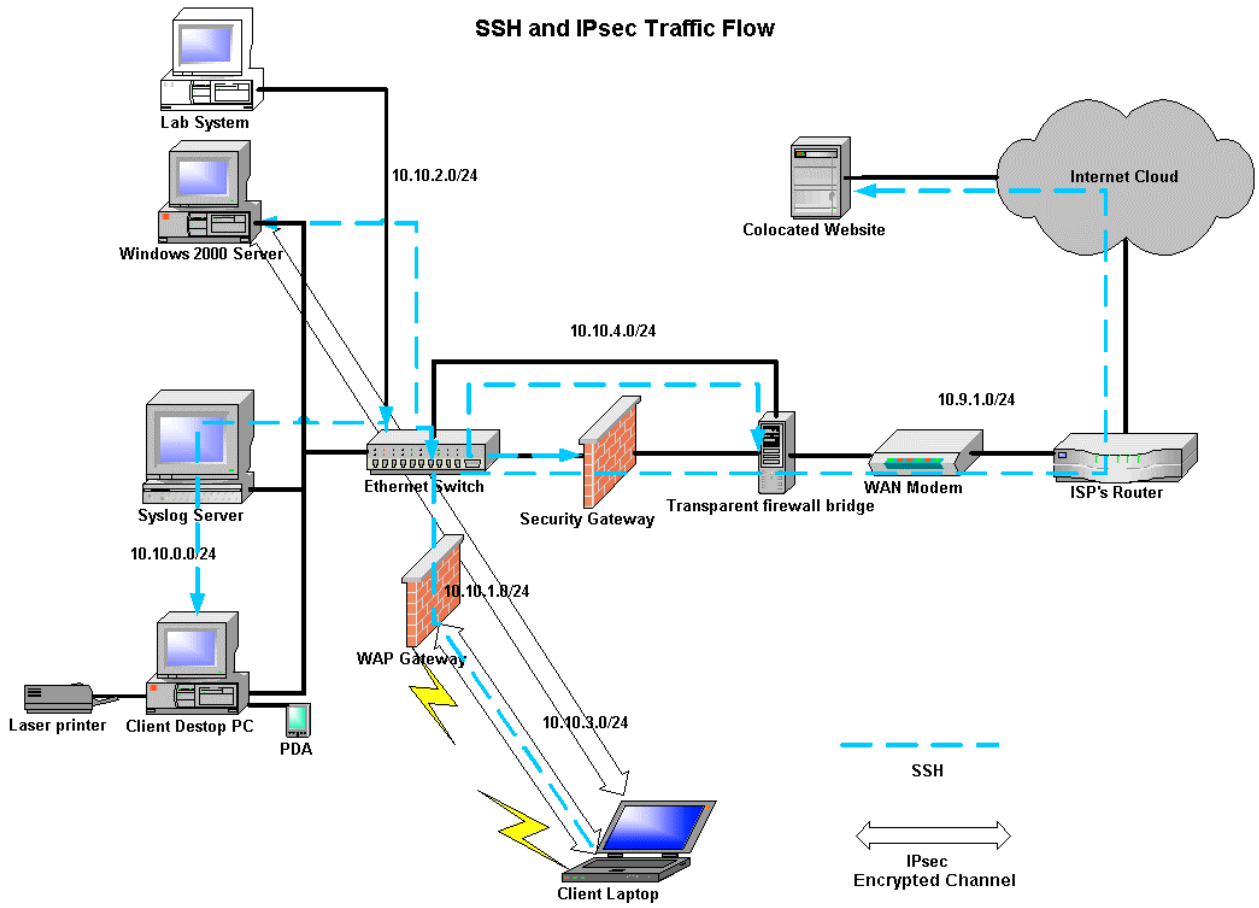

Figure 6: SSH and IPsec Traffic Flow

After building and testing the SGW prototype the plan was to swap it temporarily with the old firewall and rebuild that that system with OpenBSD 3.3. Unfortunately this did not work. OpenBSD did not recognize the disk and would not proceed with the install. After subsequent research the concept of Linux bridging emerged.[43] The decision was made to further enhance the perimeter security by using the 486 to create a bridge.

An earlier version of Debian had worked on the 486 so perhaps a later version would. The Debian r3.0 Woody installation was started and it recognized the disk like an old friend. An additional Ethernet interface was added bringing the total to 3, fortunately ISA Ethernet cards can still be found at Fry's. The reason for the third card will be explained later.

The source code for the 2.4.21 kernel was downloaded to acquire support for bridging.[44] The kernel was configured and compiled until there was a working version. Compiling the kernel on a 66Mhz 486 was real fun, I got to catch up on some reading. I elected to build all of the IP capability and the necessary Ethernet drivers into the kernel to squeeze as

[43] Bryan. "Liability Limiting." 25 Nov 2002. URL: http://honeynet.overt.org/index.php/Liability%20Limiting (5 Aug 2003).
[44] "The Linux Kernel Archives." URL: http://www.kernel.org/. (1 Aug. 2003).

much performance out of the 66Mhz 486 as possible. I added the line below to lilo.conf to get the kernel to recognize all three Ethernet interfaces.[45]

```
"append="ether=0,0,eth0 ether=0,0,eth1 ether=0,0,eth2" "
```

A bridge script was written and installed in /etc/init.d to configure and start or stop the bridge. A link was created from /etc/rc2.d/ S15bridge to the bridge script so bridging would start when the system booted. The rc2.d directory corresponds to default run level 2 specified in the inittab. S15bridge was selected so it would execute before S20snort. Snort is configure to listen on the bridge and will not start if that interface is not up. Another link was created in /etc/rd0.d and /etc/rc6.d called K10bridge so the bridge would be stopped when the system was shut down. The bridge script is in Appendix A

Snort was installed to monitor traffic passing through the bridge directed at the Internet IP address on the SGW. The third interface, mentioned above, has a private IP address for communication with the internal network through the SGW. The bridge interfaces do not have IP addresses so the third interface is required for management and data transfer.

The installation of the bridge completed the building of the perimeter defense. With the perimeter construction complete the focus turned to the network infrastructure.

Network Infrastructure

The network infrastructure meant building subnets, configuring a secure name service and creating internal firewall rulesets.

Subnetting and Name Resolution Service

The internal network is divided into segments using RFC1918 private addresses. DNS is used for name resolution. The SGW running Bind 9.2.2 is the primary DNS server for the internal network. Its responses are restricted to DNS queries from the internal network by using an acl. It is the only system allowed to send queries out to the Internet. That is accomplished by specifying the IP address of the ISP's name server as the forwarder. Zone transfers are not allowed. The SGW named.conf configuration file is in Appendix A. Instructions for configuring DNS are in DNS and Bind[46]

The WAP gateway also provides limited DNS service to authenticated wireless clients. Its main function is to forward those queries to the SGW DNS server. It was configured this way so the first hop of the DNS query, traveling over the wireless connection, would use the encrypted channel to the WAP. This is the same method used for SSH traffic mentioned above. The WAP named.conf configuration file is shown in Appendix A.

---

[45] Becker, Don. "Mini-HowTo on using multiple ethercards with Linux"
URL: http://yebisu.ics.es.osaka-u.ac.jp/linux/mini/Multiple-Ethernet.html (8 Aug. 2003)

[46] Albitz, Paul. Liu, Cricket. "DNS and Bind" 4th edition The networking CD Bookshelf Version 2.0. Apr. 2001.

Internal Firewall

The SGW doubles as the internal firewall. All of the internal subnets are attached to a switch that passes the network traffic through the SGW. PF on the SGW acts like a traffic cop and allows authorized traffic to pass to their destinations while blocking all other traffic. An examinations of the pf.conf configuration file shows that DNS traffic from all internal systems is allowed to pass to the DNS server. Web traffic is permitted to travel to the Internet and syslog traffic to the syslog server. SSH traffic can flow throughout the internal network and to the colocated website.

With carefully constructed rules traffic can be passed or blocked depending on source or destination IP address and port number. Addresses can be NATed to change the source address or redirected to modify the destination address. The authpf rules can be tailored for each user providing a fine granularity. The Packet Filter User's Guide referenced above has detailed information on PF. The configuration files are in Appendix A.

Encrypted Communications

While SSH resides on the host and perhaps has host security implication, it involves network traffic so it is included here. The Berkeley R commands, along with telnet and ftp are not allowed. They have been disabled on all systems and removed wherever possible. They are not permitted to pass through the firewall. Logins and file transfers are restricted to the SSH protocol. Review the SSH documentation for a better understanding of SSH.[47]

Host

Host Intrusion Detection System

Host intrusion detection is handled with regular system integrity monitoring. Each operating system uses different mechanisms. On the OpenBSD systems mtree along with crontab in conjunction with a runtime securelevel of 2 is used. On the Linux and Windows system tripwire is used. A regularly schedule job is run to check and report on the systems integrity.

System Hardening

Closing all unused ports and removing unnecessary services is the method used to harden all systems.

Encryption

Encryption capabilities are available for all the operating systems. The IT uses encryption to protect critical files. The owner of the company will use encryption to protect information vital to the company. User file encryption requires a security policy. Security policies are

[47] Barrett, Daniel J. Silverman, Richard E. "SSH The Secure Shell, The Definitive Guide." Sebastopol: O'Reilly & Associates, Inc. 2001.

outside the scope of this project however the tools, mechanisms, procedures and even scripts are made available and can be employed whenever a policy is put in place.

Build Methodology

The three perimeter defense systems were built off line as prototypes. Each system was plugged into the main network briefly and tested then removed. This procedure was reiterated until they were completely configured. After final testing the configurations were backup up and stored. The operating systems were re-installed and the configuration files restored. No compilers, interpreters or windowing systems were installed. File integrity checkers were run to create unblemished hashes. The hashes were stored in secure places for continued monitoring. The OpenBSD systems were brought up to security level 2. A final security assessment was conducted using the same tools used for the pre-project security assessment along with additional tools to test newly installed components.

The internal systems were hardened. Encryption utilities were installed. Host Intrusion Detection was implemented on the UNIX/Linux. The windows systems presented a more complex problem.

The Windows 2000 server would be used as a domain controller providing various services to the windows systems and be a repository for the Windows users' home directories. The Windows 98 system would be upgraded to a more secure version of Windows. This would result in a considerable expenditure of funds. It would not only be the cost of the new operating system but it would mean purchasing replacement software for all the applications resident on the Windows 98 system. Given the current budget this was cost prohibitive.

It was decided to postpone that portion of the project until an affordable plan could be developed. That plan will need to include a procedure for examining every application for continued usefulness, finding replacements and creating a migration strategy. In the interim the Windows 98 system would be made as secure as possible. Improving security for the Windows 98 system coupled with the improvements of security in general made it an acceptable risk. Risk assessment is a vital part of implementing security.

To determine the effectiveness of the new Information Technology Infrastructure Security Architecture it was necessary to perform a post-project security assessment.

Post-project Security Assessment

Various tests were run to see if snort was monitoring packets traveling through the bridge. Nmap was run with options to execute a Christmas tree, null, and syn scan. Snort produced the following output in the portscan.log file.

```
Aug 21 19:41:57 10.9.1.1:53067 -> 10.9.1.2:492 XMAS **U*P**F
Aug 21 19:42:30 10.9.1.1:41733 -> 10.9.1.2:537 FIN *******F
Aug 21 19:43:21 10.9.1.1:41734 -> 10.9.1.2:2111 NULL ********
```

As you can see, snort caught the scans.

An additional test was run called sneeze to test exploits. Snort reported the following results in the auth.log.

```
Aug 21 09:03:35 hostbridge snort[2519]: [1:615:1] SCAN Proxy attempt
[Classification: Attempted Information Leak] [Priority: 2]: {TCP}
65.128.116.178:55461 -> 10.9.1.2:1080
Aug 21 13:27:26 hostbridge snort[2519]: [1:469:1] ICMP PING NMAP
[Classification: Attempted Information Leak] [Priority: 2]: {ICMP} 1.236.117.65
-> 10.9.1.2
Aug 21 16:42:16 hostbridge snort[2519]: [1:474:1] ICMP superscan echo
[Classification: Attempted Information Leak] [Priority: 2]: {ICMP} 68.18.246.87
-> 10.9.1.2
Aug 21 00:40:45 hostbridge snort[2519]: [1:483:2] ICMP PING CyberKit 2.2
Windows [Classification: Misc activity] [Priority: 3]: {ICMP} 66.0.152.7 ->
10.9.1.2
Aug 21 05:57:19 hostbridge snort[4196]: [1:618:1] INFO - Possible Squid Scan
[Classification: Attempted Information Leak] [Priority: 2]: {TCP}
200.152.97.160:58179 -> 10.9.1.2:3128
Aug 21 15:53:16 hostbridge snort[5618]: spp_portscan: portscan status from
10.9.1.1: 10 connections across 1 hosts: TCP(10), UDP(0)
Aug 21 15:55:42 hostbridge snort[5618]: [1:472:1] ICMP redirect host
[Classification: Potentially Bad Traffic] [Priority: 2]: {ICMP} 10.9.1.1 ->
10.9.1.2
Aug 21 19:29:20 hostbridge snort[5618]: [111:10:1] spp_stream4: STEALTH
ACTIVITY (nmap XMAS scan) detection {TCP} 10.9.1.1:53067 -> 10.9.1.2:41
Aug 21 19:42:30 hostbridge snort[5618]: [111:8:1] spp_stream4: STEALTH ACTIVITY
(FIN scan) detection {TCP} 10.9.1.1:41733 -> 10.9.1.2:826
Aug 21 19:43:21 hostbridge snort[5618]: [111:9:1] spp_stream4: STEALTH ACTIVITY
(NULL scan) detection {TCP} 10.9.1.1:41734 -> 10.9.1.2:1371
```

Snort picked up and classified the attempts. It also reported the nmap scans in the auth.log file. The differences in times are a result of selecting different samples from a very large log file.

Nmap reported the following indicating all ports on the SGW are filtered.

```
nmap 10.9.1.1

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-08-21 18:05 PDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -
P0Nmap run completed -- 1 IP address (0 hosts up) scanned in 12.084 seconds

# nmap -P0 10.9.1.1

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-08-21 18:06 PDT
All 1623 scanned ports on 10.9.1.1 are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 1355.533 seconds
```

Firewalk was run and it verified that no ports were being forwarded through the firewall.

Comparing the pre and post project security assessments shows a vast improvement in security. The following is a summary of those improvements.

Summary

The previous security measures consisted of a firewall that was running an out-of-date unpatched operating system with many published vulnerabilities. There were ports open and services running on them that were not being used. There was no ability to monitor system integrity. All systems were on the same flat network. They all had open unused ports. None of the data was protected with encryption or file permissions. Programs such as telnet and ftp were available. There was no backup procedure and data was not being sent off-site.

This has been replaced with the Information Technology Infrastructure Security Architecture. There is a bridge with a Network Intrusion Detection System between the Internet and the Internet accessible system. The bridge system is running a current operating system. The firewall is now running the latest version of OpenBSD and has a Host Intrusion Detection System running on it. New functionality has been added to the network with a Wireless Access Point with that wireless traffic running over an encrypted channel. Users are required to authentication before they are allowed access to the internal network. All systems are hardened. The network is no longer flat but has subnets separated by a firewall. Data encryption is in use and there is a documented backup procedure with provisions for sending data off-site.

The security has improved immensely. The big question, is this infrastructure secure? The only infrastructure that could be considered secure would be one that is tucked away in iron vault. It could not be accessible from any out side network and would be operated by a sealed in robot programmed to be completely ethical, the programmer's high ethics are assumed. Secure yes, functional probably not. For the majority of companies that situation is neither possible nor realistic. Your answer should have been a resounding NO. It is dangerous to think otherwise. Those involved in security, the people responsible for the corporate jewels, should never, ever think that the infrastructure they are responsible for is secure. Was the project a failure? No. Security is a transient condition making the effort to secure an ITI a continuous challenge. No matter how much effort we put into setting up security we should never relax. New vulnerabilities are discovered daily. New exploits are invented constantly. New code is written with new vulnerabilities built in. Attackers are ever vigilant; you must be too.

That summary describes the completion of the first project and the scope of this paper. A second project will start that will address performance issues, include fine-tuning and implement enhancements. We can no conclude the project.

Conclusion

While this case study examined a small company with a small budget and limited resources it covered the complete spectrum of issues very similar to those faced by larger

organizations. The approaches taken here can be transferred to more complex situations and should prove useful.

This was a long journey and now it is coming to an end. Journeys like this should be learning experiences, it certainly was for this author and I hope the readers. The biggest lesson learned is that information security is an extremely complex challenge. The person taking on that challenge must face many convoluted technical issues and confront opponents who are cunning, relentless and, in some cases, extremely talented. They are numerous, usually hidden and possess the element of surprise. Accepting that challenge is not for the faint of heart but take heart, success is possible. There are many talented dedicated professionals who make up a community that believes the information they protect is sacred and should not be violated by unauthorized access.

I will end with a quote from Eric Cole when he autographed his book for me “Have fun securing your networks. Enjoy!” After all this should be fun.

References

Numerical

1. If you are not convinced that security is a necessity, there are numerous arguments presented (see… for a brief list of references.
2. Webster's Universal Encyclopedic Dictionary. Barnes & Nobel, 2002. 945.
3. Cole, Eric. Fossen, Jason. Northcutt, Stephen. Pomeranz, Hal. SANS Security Essentials with CISSP CBK Version 2.1 Volume One Sans Press, 2003.
4. Newton, Harry Newton's Telecom Dictionary. San Francisco: CMP Books, 2003. 70.
5. Angelo, Scott M. "Security Architecture Model Component Overview." 27 Nov. 2001. 3, 15 URL: http://www.sans.org/rr/securitybasics/architecture.php. (5 Apr. 2003).
6. Arconati, Nick. "One Approach to Enterprise Security Architecture." 14 Mar. 2002. 1, 2 URL: http://www.sans.org/rr/policy/approach.php. (5 Apr. 2003).
7. Tee, Chang Boon. "Building a secure Internet Data Center Network Architecture." URL: http://www.sans.org/rr/paper.php?id=73 (20 Apr. 2003).
8. Modules can be either components or elements of components. This will be relevant later when we discuss the project architecture and design.
9. Odom, Wendell. Cisco CCNA Exam #640-507 Certification Guide. Indianapolis: Cisco Press, 2000. 78.
10. Oppenheimer, Priscilla. Top-down Network Design. Indianapolis: Macmillan Technical Publishing, 1999. 121, 123.
11. Teare, Diane. Designing Cisco Networks Indianapolis: Cisco Press 1999. 90.
12. I tend to use module, component and sometimes element interchangeable. I hope this does not confuse anyone too much. Modules can be components and visa versa. Please consider the context.
13. Cole, Eric. Fossen, Jason. Northcutt, Stephen. Pomeranz, Hal. SANS Security Essentials with CISSP CBK Version 2.1 Volume One Sans Press, 2003. 294.
14. Teare, Diane. Designing Cisco Networks Indianapolis: Cisco Press 1999. 737.
15. "Dictionary.com" URL: http://dictionary.reference.com/search?q=host. (25 Apr. 2003).
16. Newton, Harry Newton's Telecom Dictionary. San Francisco: CMP Books, 2003. 67.
17. For an explanation of the difference genealogies in the meaning of gauntlet see http://www.crh.noaa.gov/library/Grammar/Gauntlet.html. (30 Mar. 2003).
18. The assessment was conducted because there is no security policy to use as the basis for an audit.
19. As it turns out, application would require a different approach. More on this later.
20. Cole, Eric. Hackers Beware. Indianapolis: New Riders Publishing, 2002.
21. McClure, Stuart. Scambray, Joel. Kurtz, George. Hacking Exposed. Berkeley: McGraw Hill/Osborn, 2003.
22. "Cheops-ng." URL: http://cheops-ng.sourceforge.net/index.php. (25 May 2003).
23. "Nessus." URL: http://www.nessus.org/. (25 May 2003).
24. "Firewalk." URL: http://www.packetfactory.net/firewalk/. (25 May 2003).
25. Postfix was installed but never fully implemented so it was never used.
26. "Security Information." 19 Aug. 2003. URL: http://www.debian.org/security/. (22 Aug. 2003).
27. Basset, Brian. URL: http://packages.debian.org/stable/net/ipmasq.html. (10 Aug 2003).

28. "John the Ripper." URL: http://www.openwall.com/john/. (1 Jun. 2003).

29. "The SANS Security Policy Project." URL: http://www.sans.org/resources/policies/#name. (24 Aug. 2003).

30. "Security." URL: http://www.openbsd.org/security.html. (1 Apr. 2003).

31. Holland, Nick. "PF: The OpenBSD Packet Filter." 29 Jun. 2003 URL: http://www.openbsd.org/faq/pf/index.html (11 Jul. 2003).

32. Pearson, Oskar. "Squid, A User's Guide." URL: http://squid-docs.sourceforge.net/latest/html/book1.html. (30 July 2003).

33. "Tripwire." URL: http://www.tripwire.org/. (10 Aug. 2003).

34. "Gnu Privacy Guard." URL: http://www.gnupg.org/. (10 Aug. 2003).

35. Holland, Nick. "PF: The OpenBSD Packet Filter." URL: http://www.openbsd.org/faq/pf/index.html. (11 Jul. 2003).

36. "Ethereal." URL: http://www.ethereal.com/. (27 Mar. 2003).

37. Stubblefield, Adam. Ioannidis, John. Rubin, Aviel D. "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP" Rev 2 21 Aug 2001.

38. URL: http://www.cs.rice.edu/%7Eastubble/wep/wep_attack.pdf.

39. Attica. Re: "ssh through squid proxy" URL: http://www.derkeiler.com/Mailing-Lists/securityfocus/Secure_Shell/2003-03/0059.html (30 Jul. 2003).

40. "PuTTY Documentation Page." URL: http://www.chiark.greenend.org.uk/~sgtatham/putty/docs.html. (15 Aug. 2003).

41. Norris, Greg. "Re: SSH through squid proxy." 15 Mar. 2003.URL: http://www.securityfocus.com/archive/121/315169 (30 Jul. 2003).

42. Padgett, Pat. "Corkscrew." URL: http://www.agroman.net/corkscrew/. (30 Jul. 2003).

43. "Tcpdump." URL: http://tcpdump.org. (20 Mar. 2003).

44. Bryan. "Liability Limiting." 25 Nov 2002. URL: http://honeynet.overt.org/index.php/Liability%20Limiting (5 Aug 2003).

45. "The Linux Kernel Archives." URL: http://www.kernel.org/. (1 Aug. 2003).

46. Becker, Don. "Mini-HowTo on using multiple ethercards with Linux" URL: http://yebisu.ics.es.osaka-u.ac.jp/linux/mini/Multiple-Ethernet.html (8 Aug. 2003).

47. Albitz, Paul. Liu, Cricket. "DNS and Bind" 4th edition The networking CD Bookshelf Version 2.0. Apr. 2001.

48. Barrett, Daniel J. Silverman, Richard E. "SSH The Secure Shell, The Definitive Guide." Sebastopol: O'Reilly & Associates, Inc. 2001.

Alphabetical

- Albitz, Paul. Liu, Cricket. "DNS and Bind" 4th edition The networking CD Bookshelf Version 2.0. Apr. 2001.
- Angelo, Scott M. "Security Architecture Model Component Overview." 27 Nov. 2001. 3, 15 URL: http://www.sans.org/rr/securitybasics/architecture.php. (5 Apr. 2003)
- Arconati, Nick. "One Approach to Enterprise Security Architecture." 14 Mar. 2002. 1, 2 URL: http://www.sans.org/rr/policy/approach.php. (5 Apr. 2003)
- As it turns out, application would require a different approach. More on this later.

- Attica. Re: "ssh through squid proxy" URL: http://www.derkeiler.com/Mailing-Lists/securityfocus/Secure_Shell/2003-03/0059.html (30 Jul. 2003).
- Barrett, Daniel J. Silverman, Richard E. "SSH The Secure Shell, The Definitive Guide." Sebastopol: O'Reilly & Associates, Inc. 2001.
- Basset, Brian. URL: http://packages.debian.org/stable/net/ipmasq.html. (10 Aug 2003).
- Becker, Don. "Mini-HowTo on using multiple ethercards with Linux" URL: http://yebisu.ics.es.osaka-u.ac.jp/linux/mini/Multiple-Ethernet.html (8 Aug. 2003)
- Bryan. "Liability Limiting." 25 Nov 2002. URL: http://honeynet.overt.org/index.php/Liability%20Limiting (5 Aug 2003).
- "Cheops-ng." URL: http://cheops-ng.sourceforge.net/index.php. (25 May 2003).
- Cole, Eric. Fossen, Jason. Northcutt, Stephen. Pomeranz, Hal. SANS Security Essentials with CISSP CBK Version 2.1 Volume One Sans Press, 2003
- "Cole, Eric. Hackers Beware. Indianapolis: New Riders Publishing, 2002
- "Dictionary.com" URL: http://dictionary.reference.com/search?q=host. (25 Apr. 2003).
- "Ethereal." URL: http://www.ethereal.com/. (27 Mar. 2003).
- "Firewalk." URL: http://www.packetfactory.net/firewalk/. (25 May 2003).
- For an explanation of the difference genealogies in the meaning of gauntlet see http://www.crh.noaa.gov/library/Grammar/Gauntlet.html. (30 Mar. 2003)
- "Gnu Privacy Guard." URL: http://www.gnupg.org/. (10 Aug. 2003).
- Holland, Nick. "PF: The OpenBSD Packet Filter." 29 Jun. 2003 URL: http://www.openbsd.org/faq/pf/index.html (11 Jul. 2003)
- I tend to use module, component and sometimes element interchangeable. I hope this does not confuse anyone to much. Modules can be components and visa versa. Please consider the context.
- If you are not convinced that security is a necessity, there are numerous arguments presented (see… for a brief list of references.
- "John the Ripper." URL: http://www.openwall.com/john/. (1 Jun. 2003).
- McClure, Stuart. Scambray, Joel. Kurtz, George. Hacking Exposed. Berkeley: McGraw Hill/Osborn, 2003.
- Modules can be either components or elements of components. This will be relevant later when we discuss the project architecture and design.
- Nessus." URL: http://www.nessus.org/. (25 May 2003).
- Newton, Harry Newton's Telecom Dictionary. San Francisco: CMP Books, 2003.
- Norris, Greg. "Re: SSH through squid proxy." 15 Mar. 2003.URL: http://www.securityfocus.com/archive/121/315169 (30 Jul. 2003)
- Odom, Wendell. Cisco CCNA Exam #640-507 Certification Guide. Indianapolis: Cisco Press, 2000. 78.
- Oppenheimer, Priscilla. Top-down Network Design. Indianapolis: Macmillan Technical Publishing, 1999. 121, 123.
- Padgett, Pat. "Corkscrew." URL: http://www.agroman.net/corkscrew/. (30 Jul. 2003).
- Pearson, Oskar. "Squid, A User's Guide." URL: http://squid-docs.sourceforge.net/latest/html/book1.html. (30 July 2003).
- Postfix was installed but never fully implemented so it was never used.

- "PuTTY Documentation Page." URL: http://www.chiark.greenend.org.uk/~sgtatham/putty/docs.html. (15 Aug. 2003).
- "Security Information." 19 Aug. 2003. URL: http://www.debian.org/security/. (22 Aug. 2003).
- "Security." URL: http://www.openbsd.org/security.html. (1 Apr. 2003).
- Stubblefield, Adam. Ioannidis, John. Rubin, Aviel D. "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP" Rev 2 21 Aug 2001.
- "Tcpdump." URL: http://tcpdump.org. (20 Mar. 2003).
- Teare, Diane. Designing Cisco Networks Indianapolis: Cisco Press 1999.
- Tee, Chang Boon. "Building a secure Internet Data Center Network Architecture." URL: http://www.sans.org/rr/paper.php?id=73 (20 Apr. 2003)
- The assessment was conducted because there is no security policy to use as the basis for an audit.
- "The Linux Kernel Archives." URL: http://www.kernel.org/. (1 Aug. 2003).
- "The SANS Security Policy Project." URL: http://www.sans.org/resources/policies/#name. (24 Aug. 2003).
- "Tripwire." URL: http://www.tripwire.org/. (10 Aug. 2003).
- URL: http://www.cs.rice.edu/%7Eastubble/wep/wep_attack.pdf
- Webster's Universal Encyclopedic Dictionary. Barnes & Nobel, 2002. 945.

Appendix A Configuration Files

Second caveat. These configuration files are included here for educational purposes only. There is no guarantee, expressed or implied, that they will work in your environment and I am not responsible for any negative consequences. I strongly recommend that you develop and test your own. You are, however, welcomed to use these as guidelines.

Security Gateway pf.conf configuration file

```
# Tables
#     Table for internal networks behind the firewall

table <GATEWAYIPS>   { 10.10.0.1, 10.10.1.1, 10.10.2.1, 10.10.4.1 }
table <INTERNALNETS> { 10.10.0.0/24, 10.10.1.0/24, 10.10.2.0/24, 10.10.4.0/24 }

# Hosts

#     WAN IP addresses

WANIPADDRESS      = "10.9.1.1"
WANGATEWAY        = "10.9.1.2"
WANNAMESVR        = "10.9.1.3"
MAILSVR           = "10.9.1.4"
SHELLSVR          = "10.9.1.4"
NTPSVR            = ""

#     LAN IP addresses
```

```
LANNAMESVR        = "10.10.0.1"
SYSLOGSVR         = "10.10.0.50"
MAILCLIENT        = "10.10.0.3"
HOSTWAPCLI4       = "10.10.3.4"
BRIDGESVR         = "10.10.4.2"

# Gateways

#     Internal gateways

SECGATEWAY        = "10.10.1.1"
WAPGATEWAY        = "10.10.1.2"
PROGATEWAY        = "10.10.0.1"
LABGATEWAY        = "10.10.2.1"
CLIGATEWAY        = "10.10.3.2"
WINGATEWAY        = "10.10.0.5"
BRGGATEWAY        = "10.10.4.1"

# Networks

PRONETWORK        = "10.10.0.0/24"
SECNETWORK        = "10.10.1.0/24"
LABNETWORK        = "10.10.2.0/24"
WAPNETWORK        = "10.10.3.0/24"
BRGNETWORK        = "10.10.4.0/24"

# Interfaces

WAN_INF           = "fxp0"
EXT_INF           = "em0"
INT_INF           = "em0"
ENC_INF           = "enc0"
PRO_INF           = "em0"
WAP_INF           = "em0"
LAB_INF           = "em0"
BRG_INF           = "em0"

ALL_INF     = "{" $WAN_INF $INT_INF $ENC_INF $WAN_INF $BRG_INF "}"
ALL_IN_INF  = "{" $PRO_INF $WAP_INF $LAB_INF $BRG_INF "}"

# Protocols

ICMP_REQUEST      = "echoreq"

# scrub all packets on entry

scrub in all
# Nat all interfaces for traffic going to the Internet

nat on $WAN_INF from { <INTERNALNETS> $WAPNETWORK } \
            to  { ! <INTERNALNETS>, !  $WAPNETWORK ! $WANIPADDRESS } -> $WANIPADDRESS

block drop log all

antispoof quick for fxp0 inet

# Pass ssh traffic to shell server
```

```
pass in log on $ALL_IN_INF proto tcp from  <INTERNALNETS>  \
      to $SHELLSVR port ssh keep state

pass out log on $WAN_INF proto tcp from  $WANIPADDRESS     \
      to $SHELLSVR port ssh keep state

# Pass icmp traffic to and from LAN

pass in log quick on $ALL_IN_INF proto icmp from { <INTERNALNETS> $WAPNETWORK }   \
      to { <INTERNALNETS> $WAPNETWORK $WANIPADDRESS } icmp-type $ICMP_REQUEST   \
        keep state

pass out log quick on $ALL_IN_INF proto icmp from { <INTERNALNETS> $WAPNETWORK    \
        $WANIPADDRESS } to { <INTERNALNETS> $WAPNETWORK $WANIPADDRESS } icmp-type \
      $ICMP_REQUEST keep state

pass in log on $ALL_IN_INF proto icmp from <GATEWAYIPS> \
      to { <INTERNALNETS> $WAPNETWORK }

pass out log on $ALL_IN_INF proto icmp from <GATEWAYIPS> \
      to { <INTERNALNETS> $WAPNETWORK }
# Pass ssh traffic around LAN

pass in log on $ALL_IN_INF proto { tcp udp } from { <INTERNALNETS> $WAPNETWORK } \
      to { <INTERNALNETS> $WAPNETWORK } port ssh keep state

pass out log on $ALL_IN_INF proto { tcp udp } from { <INTERNALNETS> $WAPNETWORK } \
      to { <INTERNALNETS> $WAPNETWORK } port ssh keep state

# Pass traffic to WAN gateway

pass out log on $WAN_INF proto { tcp udp icmp } from $WANIPADDRESS \
      to $WANGATEWAY keep state

# Pass DNS traffic between WAN and LAN nameservers

pass out log on $WAN_INF proto { tcp udp } from $WANIPADDRESS \
        to $WANNAMESVR port domain keep state

# Pass Mail traffic between LAN and LAN nameserver

pass in log on $PRO_INF proto { tcp udp } from $MAILCLIENT  \
        to $MAILSVR port = pop3s keep state

pass out log on $WAN_INF proto { tcp udp } from $WANIPADDRESS \
        to $MAILSVR port = pop3s keep state

# Pass DNS traffic between within LAN

pass in log on $ALL_IN_INF proto { tcp udp } from { <INTERNALNETS> $WAPNETWORK } \
      to <GATEWAYIPS> port domain keep state

pass out log on $ALL_IN_INF proto { tcp udp } from <GATEWAYIPS> port domain \
      to { <INTERNALNETS> $WAPNETWORK } keep state

# Pass http and https traffic from WAP network to anywhere
```

```
pass in log on $ALL_IN_INF proto { tcp udp } from { <INTERNALNETS> $WAPNETWORK }  \
        to any port { www, https } keep state

pass out log on $WAN_INF proto { tcp udp } from $WANIPADDRESS \
      to any port { www, https } keep state

# passing isakmp and encrypted traffic

pass in proto esp from { <INTERNALNETS> $WAPNETWORK } \
      to { <INTERNALNETS> $WAPNETWORK }

pass out proto esp from { <INTERNALNETS> $WAPNETWORK } \
        to { <INTERNALNETS> $WAPNETWORK }

pass in proto udp from { <INTERNALNETS> $WAPNETWORK } \
      to { <INTERNALNETS> $WAPNETWORK } port isakmp

pass out proto udp from { <INTERNALNETS> $WAPNETWORK } \
        to { <INTERNALNETS> $WAPNETWORK } port isakmp

# Passing ssh traffic between security gateways

pass in on $ENC_INF proto { tcp udp } from $WAPGATEWAY  \
      to $SECGATEWAY port ssh keep state

pass out on $ENC_INF proto { tcp udp } from $SECGATEWAY \
      to $WAPGATEWAY port ssh keep state

# Passing traffic from all internal systems to log server

pass in quick log on $ALL_IN_INF proto udp from { <INTERNALNETS> $WAPNETWORK } to
$SYSLOGSVR port syslog

pass out quick log on $PRO_INF from { <INTERNALNETS> $WAPNETWORK } to $SYSLOGSVR
```

WAP gateway pf.conf configuration file

```
# Tables

table <GATEWAYIPS>   { 10.10.0.1, 10.10.1.1, 10.10.2.1 }
table <INTERNALNETS> { 10.10.0.0/24, 10.10.1.0/24, 10.10.2.0/24 }

# LAN IP addresses

LANNAMESVR        = "10.10.1.1"
SYSLOGSVR         = "10.10.0.5"
WAPNAMESVR        = "10.10.3.2"

# Hosts

TESTLAPTOP        = "10.10.1.30"
HOSTWAPCLI4       = "10.10.3.4"
HOSTWAPCLI5       = "10.10.3.5"
```

```
# Gateways

# Internal gateways

SECGATEWAY        = "10.10.1.1"
WAPGATEWAY        = "10.10.1.2"
PROGATEWAY        = "10.10.0.1"
LABGATEWAY        = "10.10.2.1"
CLIGATEWAY        = "10.10.3.2"
WINGATEWAY        = "10.10.0.5"

# Networks

PRONETWORK        = "10.10.0.0/24"
SECNETWORK        = "10.10.1.0/24"
LABNETWORK        = "10.10.2.0/24"
WAPNETWORK        = "10.10.3.0/24"

# interfaces

WAP_INF           = "ep1"
WIFI_INF          = "an0"
ENC_INF           = "enc0"

ALL_INF           = "{" $WAP_INF $WIFI_INF $ENC_INF "}"

# Protocols

ICMP_REQUEST      = "echoreq"

scrub in all


# rulesets

#     Default deny

block drop log all

# antispoof quick for $WIFI_INF  inet

# Pass icmp traffic to and from LAN

pass in log quick on $ALL_INF proto icmp from { <INTERNALNETS> $WAPNETWORK }
\
        to { <INTERNALNETS> $WAPNETWORK } icmp-type $ICMP_REQUEST keep state
\

pass out log quick on $ALL_INF proto icmp from { <INTERNALNETS> $WAPNETWORK }
\
        to { <INTERNALNETS> $WAPNETWORK } icmp-type $ICMP_REQUEST keep state
\

# Passing in ssh traffic from security gateway

pass in quick on $WAP_INF proto { tcp udp } from $SECGATEWAY \
```

```
      to $WAPGATEWAY port ssh keep state

pass out quick on $WAP_INF proto { tcp udp } from $WAPGATEWAY \
      to $SECGATEWAY port ssh keep state
# Passing dns traffic to security gateway

pass in quick on $WAP_INF proto { tcp udp } from $WAPGATEWAY \
        to $LANNAMESVR port domain keep state

pass out quick on $WAP_INF proto { tcp udp } from $WAPGATEWAY \
      to $LANNAMESVR port domain keep state
#
# Allowing isakmp traffic

# passing in encrypted traffic from WIFI hosts

pass in log proto esp from $HOSTWAPCLI4 to $CLIGATEWAY
pass out log proto esp from $CLIGATEWAY to $HOSTWAPCLI4

pass in log proto esp from $HOSTWAPCLI4 to $WINGATEWAY
pass out log proto esp from $WINGATEWAY to $HOSTWAPCLI4

pass in log proto esp from $HOSTWAPCLI5 to $CLIGATEWAY
pass out log proto esp from $CLIGATEWAY to $HOSTWAPCLI5

# passing in traffic from designated systems on encapsulation interface

# pass in log on $ENC_INF from $HOSTWAPCLI4 to $CLIGATEWAY
# pass out log on $ENC_INF from $CLIGATEWAY to $HOSTWAPCLI4

# passing in isakmpd traffic from the WIFI network encapsulation interface

pass in log on $WIFI_INF proto udp from $HOSTWAPCLI4 port isakmp \
      to $CLIGATEWAY port isakmp

pass out log on $WIFI_INF proto udp from $CLIGATEWAY port isakmp \
      to $HOSTWAPCLI4 port isakmp

pass in log on $WIFI_INF proto udp from $HOSTWAPCLI5 port isakmp \
        to $CLIGATEWAY port isakmp

pass out log on $WIFI_INF proto udp from $CLIGATEWAY port isakmp \
        to $HOSTWAPCLI5 port isakmp

# AUTHPF
pass in quick on $ENC_INF proto ipencap all

pass out quick on $ENC_INF all

pass out quick on $WAP_INF proto tcp from $WAPNETWORK flags S/SA \
      modulate state

pass out quick on $WAP_INF proto { udp, icmp } from $WAPNETWORK \
      keep state

pass in quick on $ENC_INF proto { udp tcp } from $WAPNETWORK to $ENC_INF \
      port ssh flags S/SA keep state
```

```
anchor authpf in on $ENC_INF
anchor authpf out on $ENC_INF
anchor authpf in on $WAP_INF
anchor authpf out on $WAP_INF
```

User's authpf.rules file

```
# User authpf file

ENC_INF           = "enc0"
WAP_INF           = "ep1"

CLIGATEWAY        = "10.10.3.2"
WAPNAMESVR        = "10.10.3.2"
SQUIDPROXYOUT     = "10.10.1.2"
SQUIDPORT         = "6000"

# Pass DNS traffic from encrypted interface

pass in quick on $ENC_INF proto udp from $user_ip \
      to $WAPNAMESVR  port domain keep state

pass out quick on $ENC_INF proto udp from $WAPNAMESVR \
      to $user_ip port domain keep state

# Pass www, ssh, https from encrypted interface

pass in quick on $ENC_INF proto { udp tcp } from $user_ip \
      to $CLIGATEWAY port $SQUIDPORT flags S/SA keep state

pass in quick on $ENC_INF proto { tcp udp } from $user_ip \
      to any port { www https } flags S/SA

pass out quick on $WAP_INF proto { tcp udp } from $SQUIDPROXYOUT \
      to any port { www https } keep state

# protocols required form IPsec

pass in proto esp from $user_ip to any
pass out proto esp from $user_ip to any

pass in proto esp from any to $user_ip
pass out proto esp from any to $user_ip

pass in proto udp from $user_ip port isakmp to any port isakmp
pass out proto udp from $user_ip port isakmp to any port isakmp

pass in proto udp from any port isakmp to $user_ip port isakmp
pass out proto udp from any port isakmp to $user_ip port isakmp
```

Security Gateway named.conf configuration file

```
// BIND configuration file
```

```
options {
      directory "/";
      allow-query { internal; };
      forwarders { 10.10.9.3.3; };
      forward only;
      // Place additional options here.
};

acl "internal" {
      127/8; 10.10.0/24; 10.10.1/24; 10.10.2/24; 10.10.3/24; 10.10.4/24;
};

zone "." in {
      type hint;
      file "root.hint";
};

zone "localnet" in {
      type master;
      file "domain-info.localnet";
      allow-transfer { none; };
};

zone "0.10.10.in-addr.arpa" in {
      type master;
      file "inverse-domain-info.10.10.0";
      allow-transfer { none; };
};

zone "1.10.10.in-addr.arpa" in {
      type master;
      file "inverse-domain-info.10.10.1";
      allow-transfer { none; };
};

zone "2.10.10.in-addr.arpa" in {
      type master;
      file "inverse-domain-info.10.10.2";
      allow-transfer { none; };
};

zone "3.10.10.in-addr.arpa" in {
      type master;
      file "inverse-domain-info.10.10.3";
      allow-transfer { none; };
};

zone "4.10.10.in-addr.arpa" in {
      type master;
      file "inverse-domain-info.10.10.4";
      allow-transfer { none; };
};
```

WAP gateway named.conf configuration file

```
// BIND configuration file
```

```
options {
        directory "/";
        allow-query { wireless; };
        forwarders { 10.10.1.1; };
        forward only;
        // Place additional options here.
};

acl "wireless" {
        127/8; 10.10.3/24; 10.10.1/24;
};
```

Bridge Configuration Script

```
#!/bin/sh -e
#
#Script to configure and start or stop the bridge
#
BRGCNTL=/usr/local/sbin/brctl
IFCONFIG=/sbin/ifconfig
BRIDGE_INF=brg0
ETH_WAN_INF=eth0
ETH_LAN_INF=eth1
case "$1" in
      start)
            echo ""
            echo " *** Starting Ethernet Bridge ***"
            echo ""
            $IFCONFIG ${ETH_WAN_INF} 0.0.0.0 up
            $IFCONFIG ${ETH_WAN_INF} 0.0.0.0 up
            $BRGCNTL addbr ${BRIDGE_INF}
            $BRGCNTL addif ${BRIDGE_INF} ${ETH_WAN_INF}
            $BRGCNTL addif ${BRIDGE_INF} ${ETH_LAN_INF}
            $IFCONFIG ${BRIDGE_INF} up
            $IFCONFIG ${ETH_WAN_INF} up
            $IFCONFIG ${ETH_LAN_INF} up
            exit 0
            ;;

      stop)
            echo ""
            echo " *** Stopping Ethernet Bridge ***"
            echo ""
            $BRGCNTL delif ${BRIDGE_INF} ${ETH_WAN_INF}
            $BRGCNTL delif ${BRIDGE_INF} ${ETH_LAN_INF}
            $IFCONFIG ${BRIDGE_INF} down
            $IFCONFIG ${ETH_WAN_INF} down
            $IFCONFIG ${ETH_LAN_INF} down
            $BRGCNTL delbr ${BRIDGE_INF}
            exit 0
            ;;

      restart)
            $0 stop
            $0 start
```

```
            ;;

      *)
            echo " Usage: /etc/init.d/bridge (|start|stop|restart)"
            exit 1
            ;;

esac

exit 0
```